Amendments to the Claims

1-18. (Canceled)

19. (New) A method for processing bundled instructions through execution units of a processor, comprising the steps of:

determining a mode of operation, wherein the mode of operation comprises one of a throughput mode and a wide mode;

in the throughput mode:

fetching a first bundle of instructions from a first thread of a multiply-threaded program;

distributing the first bundle to a first cluster of the execution units for execution therethrough;

fetching a second bundle of instructions from a second thread of the program; and

distributing the second bundle to a second cluster of the execution units for execution therethrough; and

in the wide mode:

fetching a third bundle of instructions from a third thread of the program;

distributing the third bundle to the first cluster for execution therethrough;

fetching a fourth bundle of instructions from the third thread of the program; and

distributing the fourth bundle to the second cluster for execution therethrough.

20. (New) The method of claim 19, further comprising:

processing the first and third bundles within the first cluster;

processing the second and fourth bundles within the second cluster.

21. (New) The method of claim 19, further comprising the step of architecting data from the first cluster to a first register file.

22. (New) The method of claim 19, further comprising the step of architecting data from the second cluster to a second register file.

23. (New) The method of claim 19, the step of fetching the first bundle comprising decoding instructions into the first bundle.

24. (New) The method of claim 19, the step of fetching the second bundle comprising decoding instructions into the second bundle.

25. (New) The method of claim 19, the step of fetching the third bundle comprising decoding instructions into the third bundle.

26. (New) The method of claim 19, the step of fetching the fourth bundle comprising decoding instructions into the fourth bundle.

27. (New) The method of claim 19, further comprising:

bypassing data between the first cluster and the second cluster, as needed, to facilitate the processing of the third bundle through the first cluster and the fourth bundle through the second cluster.

28. (New) The method of claim 27, wherein the step of bypassing the data utilizes a latch to couple the data between the first cluster and the second cluster.

29. (New) The method of claim 19, wherein the step of determining the mode of operation comprises determining a state of a configuration bit.

30. (New) The method of claim 19, further comprising the steps of:

in the throughput mode:

fetching a fifth bundle of instructions from a fourth thread of the program;

distributing the fifth bundle to the first cluster for execution therethrough;

fetching a sixth bundle of instructions from a fifth thread of the program; and

distributing the sixth bundle to the second cluster for execution therethrough.

31. (New) A processor, comprising:

a first cluster and a second cluster, wherein each of the first cluster and the second cluster comprises a plurality of execution units, wherein each of the execution units is configured to process instructions;

a configuration bit configured to specify a mode of operation, wherein the mode of operation comprises one of a throughput mode and a wide mode; and

a thread decoder configured to group instructions of a multiply-threaded program into singly-threaded bundles and to distribute the bundles to the first cluster and the second cluster according to a state of the configuration bit;

wherein during the throughput mode, the thread decoder is configured to distribute the bundles of a first thread to the first cluster for processing, and to distribute the bundles of a second thread to the second cluster for processing; and

wherein during the wide mode, the thread decoder is configured to distribute each of the bundles of a third thread to one of the first cluster and the second cluster for processing.

32. (New) The processor of claim 31, wherein each of the first cluster and the second cluster comprises a register file.

33. (New) The processor of claim 31, further comprising a latch configured to bypass date between the first cluster and the second cluster, as needed, to facilitate the processing of the bundles of the third thread.

34. (New) The processor of claim 31, wherein during the throughput mode, the thread decoder is further configured to distribute the bundles of a fourth thread to the first cluster for processing in addition to the bundles of the first thread, and to distribute the bundles of a fifth thread to the second cluster for processing in addition to the bundles of the second thread.